

BAB IV. ANALISIS DAN PERANCANGAN SISTEM

4.1 Deskripsi Sistem

Dalam penelitian ini dibangun sebuah Aplikasi Filter Pesan Menggunakan Metode N-Gram Berbasis Telegram Bot yang mempunyai tujuan untuk memilah kata buruk dan kata alay dalam pesan yang dikirimkan. Sistem ini akan membandingkan nilai probabilitas N-Gram dari kata-kata yang tersusun dalam pesan yang dikirim dengan nilai probabilitas N-Gram dari data kata yang sudah dilatih pada sistem menggunakan persamaan profil *unigram*, jika nilai persamaan kurang dari *threshold* maka kata tersebut terindikasi kata buruk dan sistem akan mem-filternya dengan simbol.

4.2 Analisa Aktor

Sistem ini menyediakan 2 jenis aktor, Admin dan User. Admin adalah aktor yang dapat mengelola data kamus kata kasar, kata alay, dataset *tweet*, dan data pesan user serta melakukan testing filter terhadap dataset *tweet*. Untuk aktor User dapat mengirimkan pesan melalui *bot* telegram dan melihat hasil filter pesan tersebut melalui respon bot.

4.3 Analisa Kebutuhan Fungsional

Analisa kebutuhan fungsional adalah persyaratan yang berisi proses/layanan apa pun yang harus disediakan sistem nanti, termasuk bagaimana sistem harus bereaksi terhadap input tertentu dan bagaimana sistem berperilaku dalam situasi tertentu. Kebutuhan fungsional pada penelitian ini dapat dilihat pada Tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional

No.	Kebutuhan Fungsional
1.	Sistem dapat mendeteksi kata yang termasuk kata kasar dan kata alay
2.	Sistem dapat menambahkan data baru untuk kata kasar, kata alay, dan data <i>testing</i>
3.	Sistem dapat menghapus data untuk kata kasar, kata alay, dan data <i>testing</i> yang sudah ditambahkan
4.	Sistem dapat melakukan testing data pesan melalui bot telegram

5.	Sistem dapat menghapus pesan yang berhasil terdeteksi mengandung kata kasar dalam grup chat telegram melalui bot telegram
----	---

4.4 Analisa Kebutuhan Non Fungsional

Analisa kebutuhan non fungsional meliputi kebutuhan perangkat lunak (*software*) dan kebutuhan perangkat keras (*hardware*).

4.4.1. Kebutuhan Perangkat Lunak

Spesifikasi umum perangkat lunak dalam penelitian ini dapat dilihat pada Tabel 4.2.

Tabel 4.2 Kebutuhan Perangkat Lunak

No.	Nama Perangkat Lunak
1.	Sistem Operasi Windows 7/8/10
2.	Teks Editor Visual Studio Code
3.	Python
4.	Browser Chrome
5.	XAMPP
6.	Telegram

4.4.2. Kebutuhan Perangkat Keras

Spesifikasi umum perangkat keras dalam penelitian ini dapat dilihat pada Tabel 4.3.

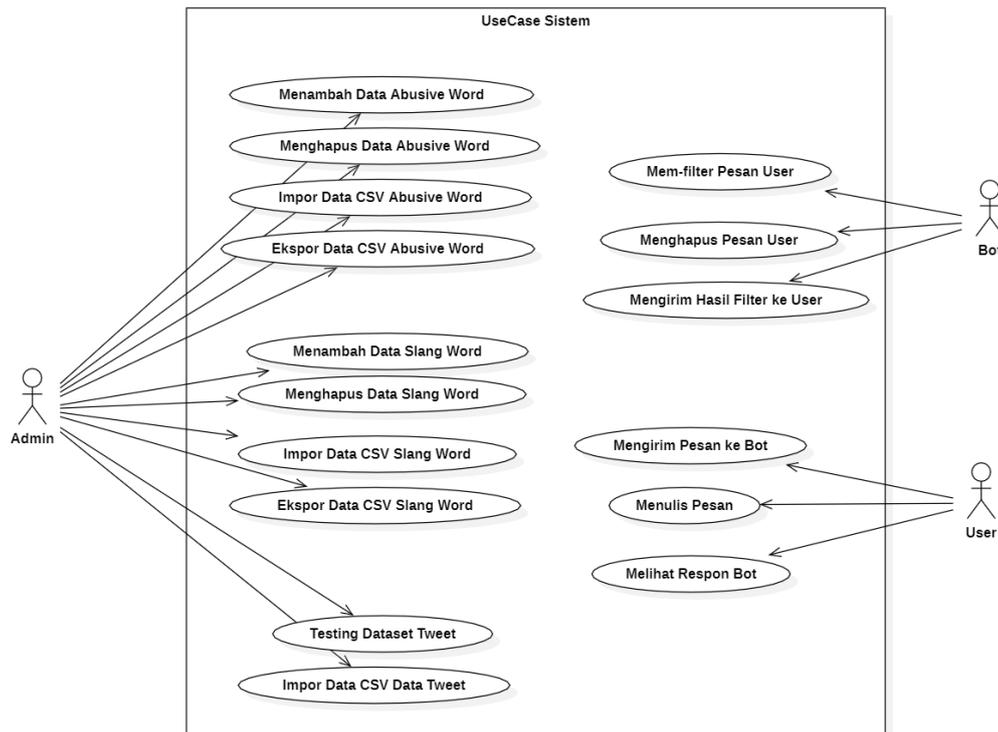
Tabel 4.3 Kebutuhan Perangkat Keras

No.	Nama Perangkat Keras
1.	Processor Intel Core i7-8750H
2.	RAM 8 GB
3.	HDD 458 GB

4.5 Perancangan Sistem

4.5.1. Diagram *Use Case*

Pada penelitian ini terdapat 2 tipe aktor, yaitu Pengguna (*user*) dan Admin. *Use case* diagram dibuat untuk mendeskripsikan interaksi antara aktor dengan sistem yang dibuat. Berikut gambaran *use case* dari penelitian ini.

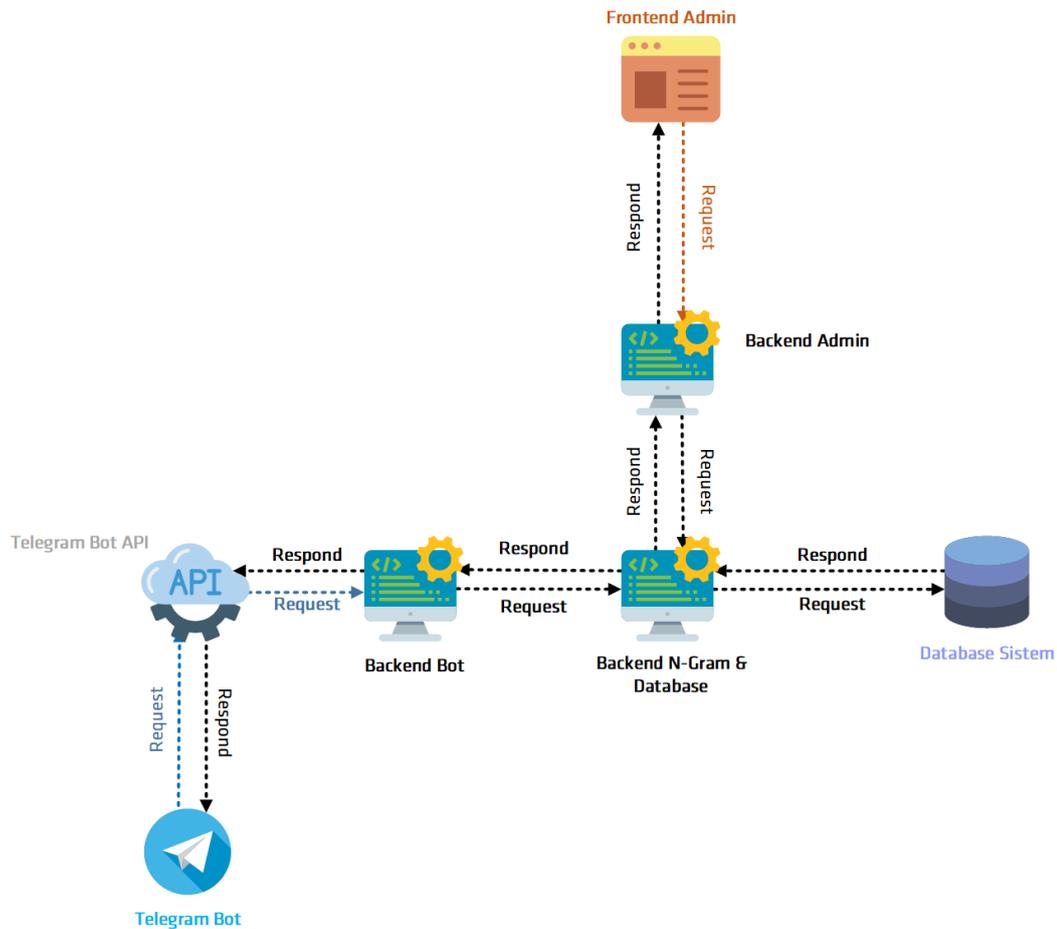


Gambar 4.1 *Use Case* Sistem

Use case di atas menunjukkan suatu tugas yang dapat dikerjakan oleh admin seperti menambah, menghapus, mengimpor csv, dan mengekspor csv dari data kata kasar (*abusive word*), data kata alay (*slang word*) serta data *tweet*. Selain itu admin dapat melakukan testing data dengan menggunakan ketiga data sebelumnya sebagai data olahnya.

Pada *use case* tersebut juga menggambarkan bahwa pengguna dapat menulis pesan dan mengirimkan pesan tersebut ke *bot* telegram dimana pesan tersebut akan dikirim ke server bot. Pengguna juga dapat melihat respon dari *bot* yang tidak lain adalah hasil *filter* pesan yang dikirimkan. Di sisi lain bot akan mem-*filter* pesan pengguna yang terindikasi kata kasar dan bot dapat menghapus pesan tersebut dengan mengirimkan hasil *filter* pesan kepada pengguna.

4.5.2. Arsitektur Sistem



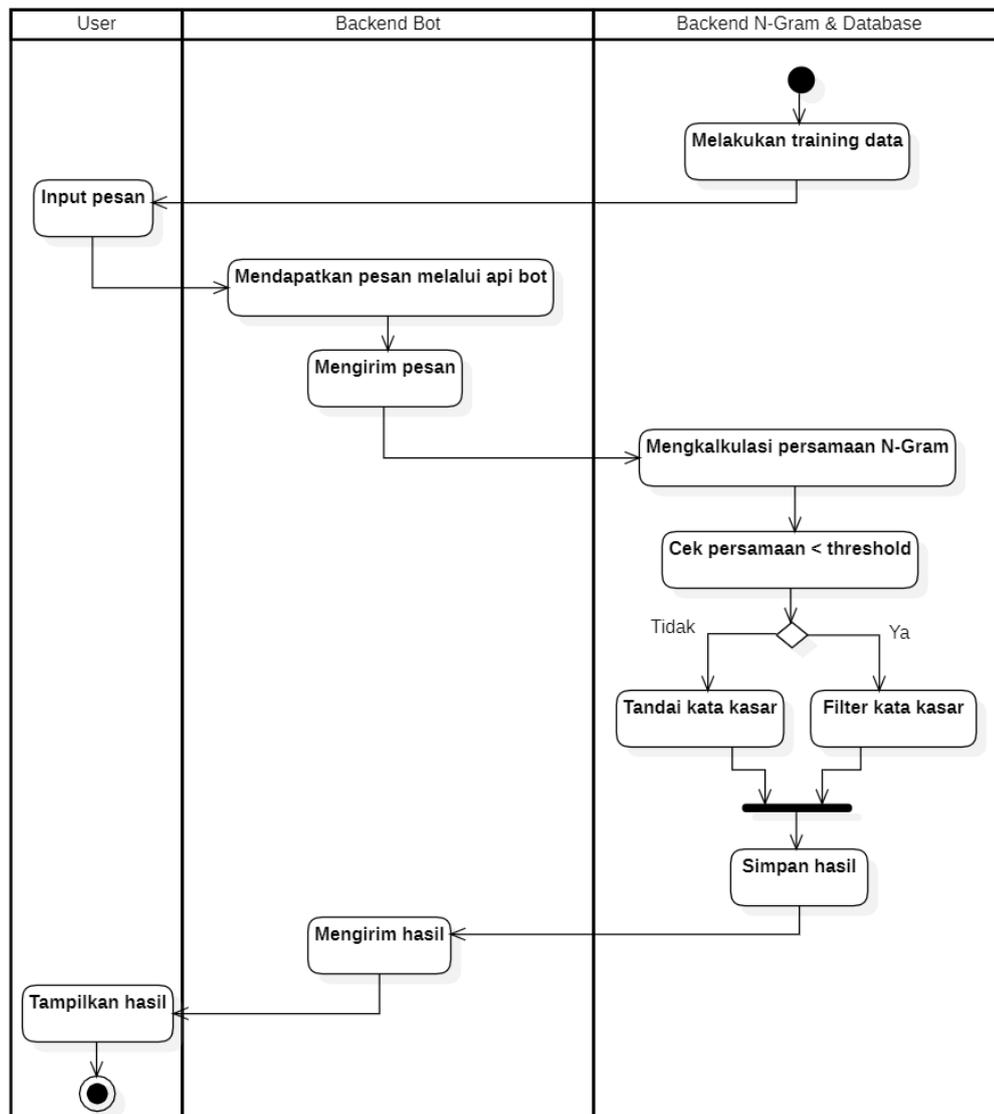
Gambar 4.2 Arsitektur Diagram

Pada arsitektur sistem di atas menggambarkan bahwa sistem ini memiliki 2 bagian, pertama bagian telegram bot dimana pengguna (*user*) yang berperan dan yang kedua bagian website frontend yang ditujukan untuk admin. Pada bagian telegram bot, sistem akan dihubungkan dengan API milik telegram bot tersebut agar dapat saling beroperasi. Backend Bot memiliki tugas untuk mengelola setiap data pesan yang dikirim user dan mengembalikan hasil filter pada user melalui respon bot. Untuk Backend Admin memiliki tugas untuk mengelola setiap data kamus yang ditambahkan oleh admin baik itu kamus kata kasar maupun kata alay serta dataset *tweet* ke dalam database sebagai bahan pengolahan data. Kedua backend tersebut terhubung dengan backend ketiga, yaitu backend N-Gram & Database. Backend ini bertugas untuk melakukan pengecekan persamaan profil *unigram* dari data pesan ataupun data *tweet* dengan data yang tersedia dalam database dan

selanjutnya akan dilakukan pemfilteran ketika nilai persamaan dengan *threshold* terpenuhi.

4.5.3. Activity Diagram Filter Pesan

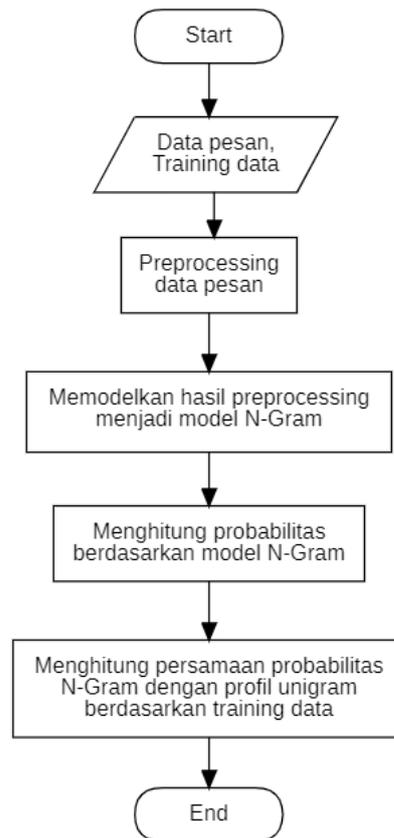
Activity diagram adalah diagram yang dapat memodelkan proses-proses yang terjadi pada sistem.



Gambar 4.3 Activity Diagram Filter Pesan

Pada Gambar 4.3 menjelaskan mengenai alur proses sistem filter pesan pada bot telegram. Awal mulai sistem berjalan akan dilakukan *training data* terlebih dahulu. Setelah sistem telah beroperasi secara keseluruhan, dilanjutkan dengan *user* yang mengirimkan pesan melalui bot. Pesan tersebut akan ditangkap oleh *backend*

bot dan dikirim ke *backend* N-Gram untuk dilakukan proses kalkulasi persamaan N-Gram-nya dan dicek dengan *threshold* hasilnya. Jika nilainya kurang dari *threshold* maka akan diteruskan dengan pemfilteran kata, sebaliknya jika lebih maka kata tersebut hanya akan ditandai saja dengan *markup*. Setelah itu hasil terakhir akan disimpan dan dikirim kembali kepada user melalui respon bot.



Gambar 4.4 Proses pada kalkulasi persamaan *N-Gram*

Data pesan yang telah diterima nantinya akan dilakukan penyaringan terlebih dahulu agar data menjadi lebih terstruktur. Proses ini dilakukan dalam preprocessing, dimana tahapan-tahapan yang dilalui sama seperti yang telah digambarkan pada Gambar 3.1.

Setelah melewati penyaringan, data akan dimodelkan dalam bentuk N-Gram yang mana pada penelitian ini digunakan model *Bigram* dan *Trigram*, namun untuk penerapannya hanya salah satu model saja yang digunakan. Lalu dari model tersebut dihitunglah probabilitas setiap kata yang ada pada pesan sesuai dengan model N-Gram yang dipilih. Setelah semua kata mendapat nilai probabilitasnya,

waktunya mengecek persamaan antara probabilitas N-Gram pesan dengan data *training* menggunakan rumus profil unigram.

4.5.4. Database

4.5.4.1. Struktur Tabel Database

Struktur tabel *database* adalah pengaturan tabel yang akan digunakan untuk menyimpan data dalam database. Struktur tabel memudahkan untuk memasukkan data sesuai dengan tipe data yang ditentukan. Struktur tabel dari penelitian ini dapat dilihat di bawah ini.

Tabel abusive adalah tabel yang digunakan untuk menyimpan data kata kasar (*abusive*). Tabel ini terdiri dari kolom *id_vocab* dengan tipe data *integer*, *word* dengan tipe data *varchar*, dan *label_vocab* dengan tipe data *integer* dimana *id_vocab* sebagai primary key.

Tabel 4.4 Deskripsi Tabel abusive

Nama Atribut	Tipe Data	Keterangan
<i>id_vocab</i>	INT(11)	PRIMARY KEY, AUTO INCREMENT
<i>word</i>	VARCHAR(255)	-
<i>label_vocab</i>	INT(11)	-

Tabel slangword adalah tabel yang digunakan untuk menyimpan data kata alay (*slangword*) dan bentuk bakunya. Tabel ini terdiri dari kolom *id_slang* dengan tipe data *integer*, *slang_word* dengan tipe data *varchar*, dan *standard_word* dengan tipe data *varchar* dimana *id_slang* sebagai primary key.

Tabel 4.5 Deskripsi Tabel slangword

Nama Atribut	Tipe Data	Keterangan
<i>id_slang</i>	INT(11)	PRIMARY KEY, AUTO INCREMENT
<i>slang_word</i>	VARCHAR(500)	-
<i>standard_word</i>	VARCHAR(500)	-

Tabel dataset adalah tabel yang digunakan untuk menyimpan data *tweet* yang telah didapat. Tabel ini terdiri dari kolom *id_dataset* dengan tipe data *integer*, *dataset* dengan tipe data *varchar*, dan *label_dataset* dengan tipe data *integer* dimana *id_dataset* sebagai primary key.

Tabel 4.6 Deskripsi Tabel dataset

Nama Atribut	Tipe Data	Keterangan
<i>id_dataset</i>	INT(11)	PRIMARY KEY, AUTO INCREMENT
<i>dataset</i>	VARCHAR(1000)	-
<i>label_dataset</i>	INT(11)	-

Tabel message adalah tabel yang digunakan untuk menyimpan data pesan yang dikirim melalui user yang terindikasi mengandung kata kasar. Tabel ini terdiri dari kolom *id_msg* dengan tipe data *integer*, *msg* dengan tipe data *varchar* dan *user_id* dengan tipe data *integer* dimana *id_msg* sebagai primary key.

Tabel 4.7 Deskripsi Tabel message

Nama Atribut	Tipe Data	Keterangan
<i>id_msg</i>	INT(11)	PRIMARY KEY, AUTO INCREMENT
<i>msg</i>	VARCHAR(1000)	-
<i>user_id</i>	INT(11)	-

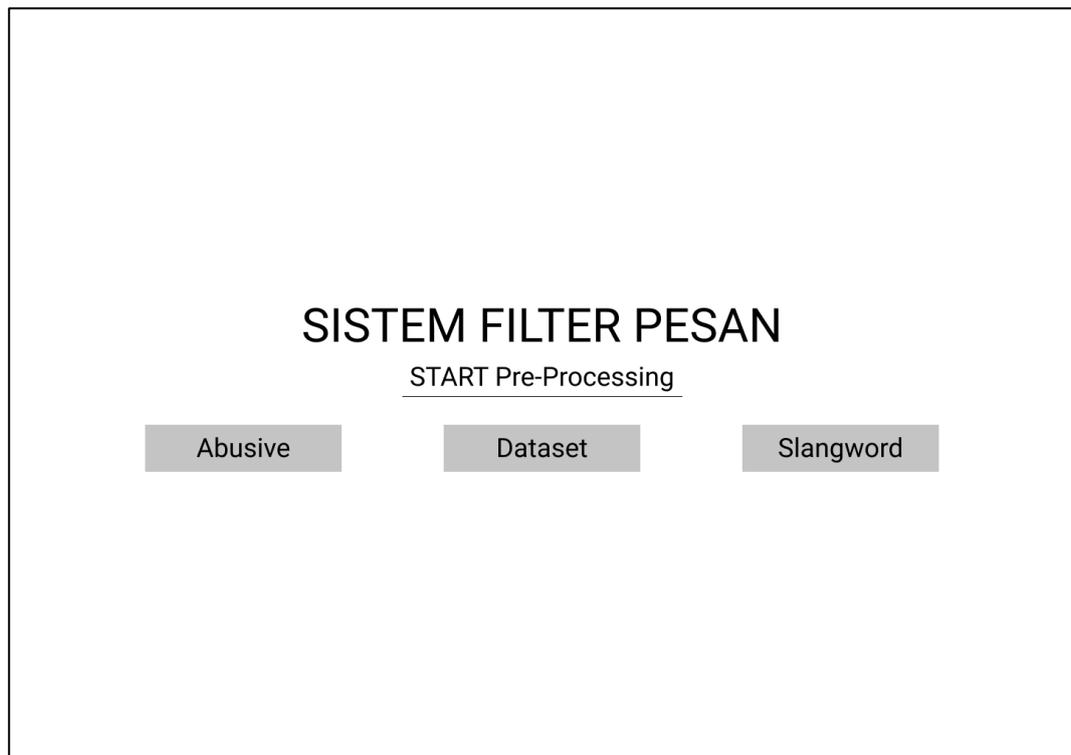
4.5.5. Desain *Interface*

Desain *interface* adalah suatu proses pembuatan tampilan dalam aplikasi dengan penekanan pada tampilan dan gaya yang bertujuan untuk memudahkan penggunaan aplikasi tersebut. Adapun desain *interface* dari penelitian ini dapat dilihat pada gambar di bawah ini.

1. Tampilan Halaman Utama

Pada Gambar 4.5 ditunjukkan rancangan antarmuka halaman utama yang dapat diakses oleh admin. Pada halaman ini terdapat *Button Abusive* dengan fungsi untuk berpindah ke halaman kamus kata kasar, *Button Dataset* untuk berpindah ke halaman dataset *tweet*, dan *Button Slangword* untuk berpindah ke halaman

kamus kata alay. Teks di bawah *title* merupakan *link* yang digunakan untuk berpindah ke halaman *testing*.



Gambar 4.5 Tampilan Halaman Utama

2. Tampilan Halaman Kamus Kata Kasar (*Abusive*)

Pada Gambar 4.6 ditunjukkan rancangan antarmuka halaman kamus kata kasar yang dapat diakses oleh admin. Pada halaman ini terdapat *form input* untuk kata kasar dan *form select* untuk memilih label dari kata tersebut. Adapun *Button Submit* dengan fungsi untuk men-*submit* form di atas, *Button Import* untuk mengimpor file csv kamus kata kasar yang didapat ke dalam *database*, *Button Export* untuk mengeksport kamus kata kasar yang tersimpan dalam *database* ke dalam file csv, dan *Button Back* digunakan untuk kembali ke halaman utama. Pada halaman ini juga ditampilkan data-data yang telah tersimpan di *database*. Terdapat *form input* untuk mencari data kata kasar tertentu dengan *Button Search* sebagai *submit form*-nya.

TAMBAH DATA ABUSIVE

DATA

LABEL

LIST DATA

CARI

Gambar 4.6 Tampilan Halaman Kamus Kata Kasar (*Abusive*)

3. Tampilan Halaman Kamus Kata Alay (*Slangword*)

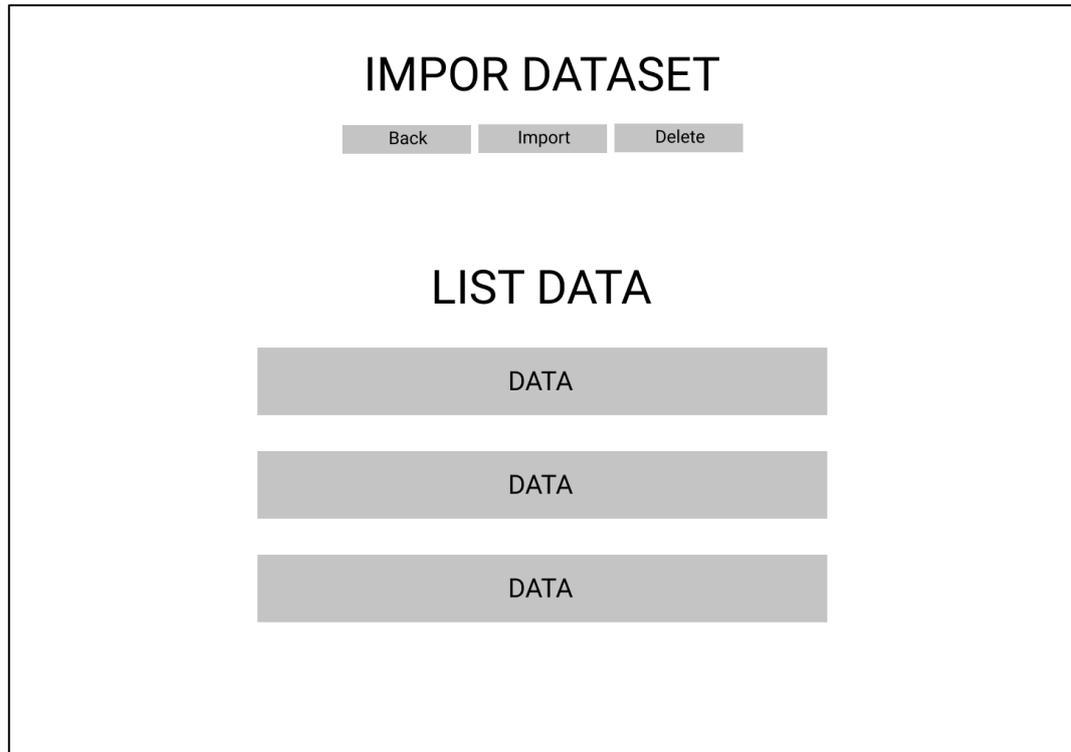
Pada Gambar 4.7 ditunjukkan rancangan antarmuka halaman kamus kata alay yang dapat diakses oleh admin. Pada halaman ini terdapat *form input* untuk kata alay dan bentuk kata bakunya. Adapun *Button Submit* dengan fungsi untuk men-submit form di atas, *Button Import* untuk mengimpor file csv kamus kata alay yang didapat ke dalam *database*, *Button Export* untuk mengeksport kamus kata alay yang tersimpan dalam *database* ke dalam file csv, dan *Button Back* digunakan untuk kembali ke halaman utama. Pada halaman ini juga ditampilkan data-data yang telah tersimpan di *database*. Terdapat form input untuk mencari data kata alay tertentu dengan *Button Search* sebagai submit form-nya.

The image shows a web application interface with two main sections. The top section is titled "TAMBAH DATA SLANG" and contains two input fields labeled "SLANG" and "BAKU". Below these fields are four buttons: "Back", "Submit", "Import", and "Export". The bottom section is titled "LIST DATA" and contains a search input field labeled "CARI" with a "Search" button. Below the search field are four buttons, each labeled "DATA".

Gambar 4.7 Tampilan Halaman Kamus Kata Alay (*Slangword*)

4. Tampilan Halaman Dataset *Tweet*

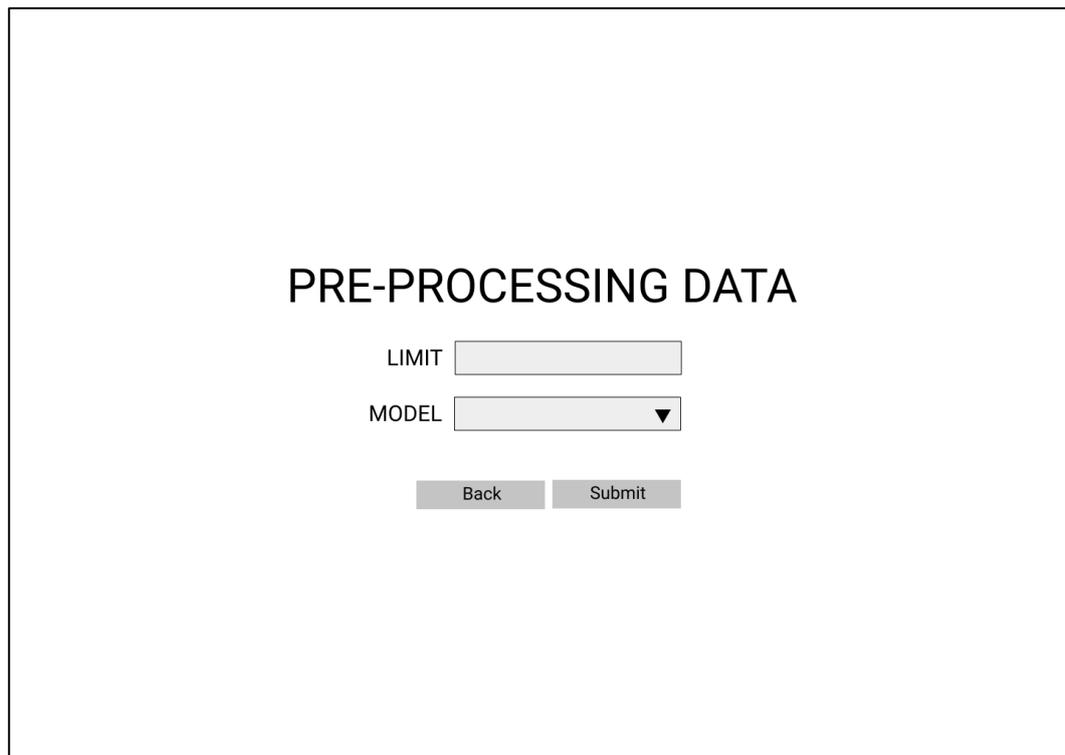
Pada Gambar 4.8 ditunjukkan rancangan antarmuka halaman dataset tweet yang dapat diakses oleh admin. Pada halaman ini terdapat *Button Import* untuk mengimpor file csv data *tweet* yang didapat ke dalam *database*, *Button Delete* untuk menghapus seluruh data *tweet* yang tersimpan dalam *database*, dan *Button Back* digunakan untuk kembali ke halaman utama. Pada halaman ini juga ditampilkan data-data yang telah tersimpan di *database*.



Gambar 4.8 Tampilan Halaman Dataset

5. Tampilan Halaman *Testing*

Pada Gambar 4.9 ditunjukkan rancangan antarmuka halaman *testing* dataset yang dapat diakses oleh admin. Pada halaman ini terdapat *form input* untuk menginputkan parameter nilai *threshold* yang akan diuji dan *form select* untuk memilih model N-Gram yang akan digunakan. Adapun *Button Submit* untuk men-submit *form* di atas, dan *Button Back* digunakan untuk kembali ke halaman utama.



PRE-PROCESSING DATA

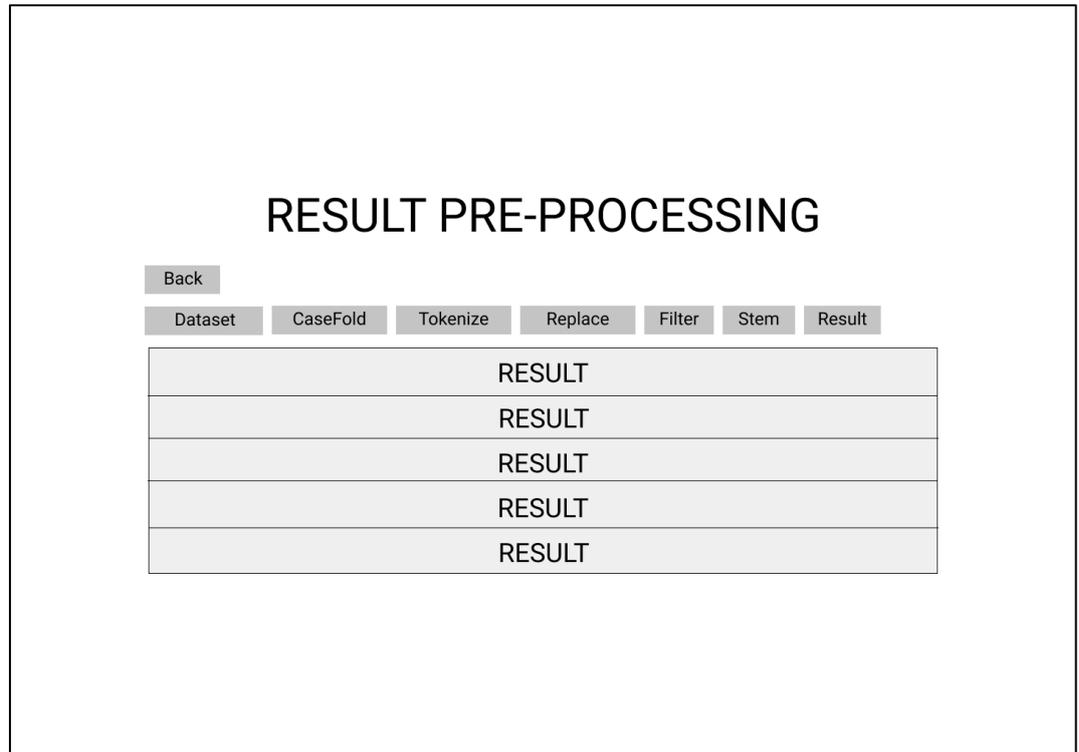
LIMIT

MODEL

Gambar 4.9 Tampilan Halaman *Testing*

6. Tampilan Halaman *Result Testing*

Pada Gambar 4.10 ditunjukkan rancangan antarmuka halaman *result testing* yang dapat diakses oleh admin. Pada halaman ini terdapat beberapa tab yang menampilkan proses dari *preprocessing* seperti tab dataset yang menampilkan bentuk dataset awal, tab *case folding* yang menampilkan proses *case folding*, tab *tokenizing*, tab *replacing*, tab *filtering*, tab *stemming*, dan tab hasil klasifikasi. Adapun *Button1* dengan fungsi untuk kembali ke halaman utama.



Gambar 4.10 Tampilan Halaman *Result Testing*